MCGINN & GIBB, PLLC
A PROFESSIONAL LIMITED LIABILITY COMPANY
PATENTS, TRADEMARKS, COPYRIGHTS, AND INTELLECTUAL PROPERTY LAW
8321 OLD COURTHOUSE ROAD, SUITE 200
VIENNA, VIRGINIA 22182-3817
TELEPHONE (703) 761-4100
FACSIMILE (703) 761-2375

APPLICATION FOR UNITED STATES LETTERS PATENT

APPLICANT: Blaum et al.

FOR: METHOD AND SYSTEM FOR SYNCHRONIZATION IN THE PRESENCE OF THERMAL ASPERITY

DOCKET NO.: ARC920000071US1

METHOD AND SYSTEM FOR SYNCHRONIZATION IN THE PRESENCE OF THERMAL ASPERITY

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention generally relates to a disk drive, and more particularly to a disk drive with dual synchronization fields in which a second synchronization field contains only a byte synchronization portion, is error tolerant and is preceded by random data.

Description of the Related Art

Oftentimes, problems may occur which destroy data in a data stream, or prevents it from being recovered.

For instance, one such problem is the so called "thermal-asperity problem". If, for instance, a thermal-asperity hits the data portion of a sector, then normally the lost data is recovered using known techniques like error-correction codes (ECC). If the thermal-asperity, on the other hand, hits the synchronization field preceding the data, then the data may be lost since, if synchronization is lost,

the proper timing or byte synchronization may not be present. In order to avoid such problems, some conventional disk drive systems utilize dual synchronization (hereafter "sync") fields for each data header. Each sync field consists of two parts: a bit-synchronization part, or variable frequency oscillator (VFO) part, and a byte synchronization part ("sync" herein after). The VFO can be thought as a preamble of 1s, while the byte sync part is a vector following it and generally possessing error tolerant properties. By utilizing two sync fields as opposed to one, the problem of catastrophic failures (such as thermal asperity) affecting it is mitigated. In effect, if the two sync fields are sufficiently separated, it is expected that at least one of them will survive a thermal-asperity type of defect. However, this approach is problematic because it takes up valuable "real estate" in the stream which otherwise could be used for the data payload. That is, this conventional method, as further described below, is disadvantageous since it limits the amount of data which could be written to the disk since such a preamble of 1s (e.g., non data) must be employed.

Hence, before reading a data sector, it is necessary to achieve correct synchronization. There are two types of synchronization that must be achieved: bit-synchronization and byte-synchronization.

To this end, as mentioned above, a synchronization field is added in front of the data sector, and this synchronization field is divided into two portions. A first portion is termed a "bit-synchronization field" (e.g., also called the preamble

or the VFO field), which normally includes a sufficiently long (i.e., its length depends on the application and the type of errors normally expected to be encountered) string of 1s. Figure 1 shows a typical example of the way double sync fields are implemented today: a first sync field, consisting of a VFO portion (denoted Sync1) followed by a byte sync portion (denoted SB1), and a second sync field, consisting of a VFO portion (denoted Sync2) followed by a byte sync portion (denoted SB2). For the sake of example, we assume that Sync1 is 24 bytes (i.e., 192 bits) long, Sync2 is 16 bytes long, and both SB1 and SB2 are 4 bytes long.

Traditionally, without the double sync field, the byte synchronization portion following the VFO is generally tolerant to a limited number of errors, and once the end of this byte synchronization portion is identified, the beginning of the data can be read.

However, as mentioned above, one of the problems with this scheme is the presence of Thermal Asperities (TA). TA occurs when, for example, the read/write head is too close to the disk being read or written to. The effect of TA is a relatively long burst of errors. For example, such errors may take the form of an "empty signal" which carries no information, such as by having a loss of amplitude of the signal (e.g., no highs or lows, thereby not allowing 1s or 0s to be detected), etc. If TA occurs in data, it is expected that the errors will be recovered by the error-correcting code (ECC) protecting the data (normally, the ECC adds

redundancy at the end of the data which is used for correction and detection of errors). Typically, Reed-Solomon (RS) codes are used as the ECC. Techniques like interleaving of RS codes permit the correction of relatively long bursts of errors.

As mentioned above, the problem changes when TA affects the sync field. In that case, a TA may cause a loss of either bit-sync or of byte-sync. Therefore, if the beginning of the data cannot be identified, then it cannot be determined where the bytes are.

To solve this problem, for instance, in one conventional system employing a disk system and a Lucent® Channel for reading data, a "double-sync" scheme has been implemented. Schematically, the double-sync in such a system appears as shown in Figure 1 (i.e., the units are in bytes).

In Figure 1, "Sync" denotes the VFO field and "SB" denotes the byte synchronization field. As shown in Figure 1, the conventional arrangement requires 24 bytes for Sync1, 4 bytes for SB1, 16 bytes for Sync2, and 4 bytes for SB2.

Hence, in the conventional system, the Lucent channel requires a 4-byte byte sync field, although simulations by the present inventors show that very good byte sync can be achieved with only two bytes of byte sync field. This should be considered for future products not using the Lucent® channel, or even if the Lucent® channel is continued, its specifications might be modified in order to

accommodate a more efficient format. Notwithstanding, there are still problems with this arrangement.

That is, in the conventional systems, a dual sync is used including a second VFO field Sync2. As a result, bit-sync can occur even in the event of TA. Further, if TA wipes out the first sync field and part of the data, the second sync field SB2 retrieves byte-sync. Obviously, there may be other causes other than TA for wiping out the byte synchronization symbol of the first field.

However, this comes at a high price of disk "real estate" being unavailable for data (e.g., payload). The highest price comes from Sync2: it uses 16 bytes, and it would be desirable to eliminate this field, provided that TAs can be handled.

SUMMARY OF THE INVENTION

In view of the foregoing and other problems, drawbacks and disadvantages of the conventional methods and structures, an object of the present invention is to provide a data structure and method for disk drive systems.

Another object is to provide dual sync bytes for each data header of a disk drive, in which the second sync byte is error tolerant but is preceded by random data, not a conventional VFO field as in conventional systems.

In a first aspect of the present invention, a method (as well as system and programmable storage medium for storing program steps of the method) of

finding byte synchronization, includes appending a synchronization symbol to random data.

With the unique and unobvious aspects of the present invention, a scheme is provided in which more space on the disk can be used for data such that dual byte sync portions can be provided for each data header of a disk drive, in which the second byte sync portion is error tolerant but is preceded by random data. For example, by using the invention, the 16 bytes corresponding to Sync2 in Figure 1 may be used for storing data.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, aspects and advantages will be better understood from the following detailed description of preferred embodiments of the invention with reference to the drawings, in which:

Figure 1 is a schematic diagram of a conventional double-sync arrangement;

Figure 2 is a perspective view representation of a disk drive data storage device for use with the present invention;

Figure 3 is a block diagram of a conventional ECC byte synchronization detector circuit which can be used with the present invention;

Figure 4 is a schematic indicating an arrangement according to the present invention;

Figure 5 illustrates a flow diagram illustrating a preferred method of the invention;

Figure 6 illustrates an exemplary hardware/information handling system 600 for incorporating the present invention therein; and

Figure 7 illustrates a signal bearing medium 700 (e.g., storage medium) for storing steps of a program of a method according to the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS OF THE INVENTION

Referring now to the drawings, and more particularly to figures 2-7, there is shown a preferred embodiment of a synchronization method (and system) in the presence of thermal asperity, and data structure for use therewith, according to the present invention.

Referring to Figure 2, an exemplary disk recording and reproducing system 200 is illustrated that can be used with the present invention. The exemplary disk drive system 200 typically contains an actuator, a servo system,

etc. for reliably reading and writing to a disk. The system 200 may include a voice coil motor (VCM) 201 for reading from/writing to a disk 202.

As shown, a feedback loop includes a driver 203A for driving the VCM 201. The head position of the VCM 201 is controlled by the feedback loop including a summing circuit 203B receiving a track position signal, a demodulator 204 for producing a position error signal, an analog-to-digital converter (ADC) 205 for converting the analog position error signal to a digital signal, a controller 206 for controlling the filter based on an input from the ADC 205, a digital filter 207, and a digital-to-analog converter 208 for providing the signal to the driver 203A.

Figure 3 illustrates a conventional ECC byte synchronization detector circuit 310, as described in U.S. Patent No. 6,089,749 to Blaum et al., which describes conventional byte sync with error correcting capabilities and incorporated herein by reference, and which can be used with the disk recording and reproducing system of the present invention.

Specifically, Figure 3 shows a schematic block diagram of an ECC byte synchronization detector circuit 310. Byte synchronization detector circuit 310 includes a vector subtractor circuit 320, an offset adder circuit 330, a distance decoder circuit 311 and a code detector circuit 312. Vector subtractor circuit 320 and offset adder circuit 330 form the primary portion of circuit 310, and provide vector subtraction and addition of the elements of the ECC code for obtaining the

Hamming distance between a zero vector and an error vector. A bit counter 313 and a window generator 314 operate in a well-known manner for producing a detecting window during the period of time when the ECC code is expected to occur.

With the above structure and environment in mind and turning now to the present invention generally, as mentioned above, the invention is directed to a method for byte synchronization in the presence of anomalous conditions (e.g., thermal asperity (TA)). It avoids the dual sync field by synchronizing directly over data without a second bit synchronization field. Hereinbelow, a method and apparatus for achieving such synchronization over data, are described.

The purpose of the invention is to provide an alternative to dual sync, basically by eliminating the second VFO field Synch2 and replacing it with data. The new scheme operates as shown in Figure 4.

That is, the structure as shown in Figure 4 provides bit-sync even in the event of TA. Perhaps, depending on the length of TAs, the VFO field Sync1 may need to be lengthened somehow, but efficiency in format is achieved.

Hence if TA destroys the first byte sync field SB1 and a portion of the data, the second byte sync field SB2 retrieves byte-sync. The difficulty is synchronizing over random data, since there is no control over it. In effect, it is possible that the second sync field appears in random data. Hereinbelow, with the

unique and unobvious structure of the present invention, it is described how to overcome these difficulties.

The inventive method will be illustrated with an exemplary 1-byte (8 bits) byte-sync pattern. This pattern maximizes the minimum distance when slided over itself. Later, a 4-byte (32 bits) pattern will be presented that is optimal with this property.

In figure 5, x represents random data, that are followed by the 8-bit pattern 0 0 0 1 1 0 1 0, and then again by random data. A window 4 bits before the pattern arrives is opened, and the window is closed 4 bits later.

Then, the inventive method measures the Hamming-distance between the pattern and the read sequence, where the distance to a data bit x counts as 0. Measuring the Hamming-distance is well known in the industry and is described, for example, in U.S. Patent No. 5,999,110 and U.S. Patent No. 6,089,749, both incorporated herein by reference.

The result is registered in the left column. Then, the window is slided until 4 bits after the end of the inventive pattern. The result is as shown in Figure 5. It can be seen that the Hamming distance is 3 or more, except when the pattern has been found. This property allows for tolerance of one error within the window considered.

The method of the present invention can be utilized with any length of byte-sync. In the preferred embodiment, a 4 byte (32 bits) byte-sync pattern is

selected. In this case, by opening the window 2 bytes before the pattern and closing it 2 bytes after the pattern, the minimum distance is 12 (i.e., it can tolerate up to 5 errors and detect 6). Since, in a partial response channel with a precoder, errors tend to occur in pairs, the pattern was utilized for correction of 4 errors only.

The 4-byte pattern chosen is

00000001 10110111 11101010 00011000

The minimum distance of 12 given by this pattern is optimal (i.e., there is no 32-bit pattern giving a distance of 13).

The system is very robust if it can be assured that the search opens in a window 2 bytes before and 2 bytes after the correct pattern is located. Unfortunately, this appears not to be the case, and the window may open up to 8 bytes before and 8 bytes after the correct pattern. Therefore, it can be looked for in random data. It is noted that "opening the window" is another common procedure in the art and would be easily known by one of ordinary skill in the art within the purview of the present application.

However, it is unlikely that this 32-bit pattern will be found in random data. If it is found, there are several possible strategies.

One strategy is to make a few artificial errors in the data to make the pattern disappear. These few errors will be later corrected by the powerful ECC.

Another alternative is to register the occurrence of the pattern in data, in which case it will be known that it has occurred previously to the correct sync field.

The system was simulated assuming 15 bytes of random data followed by the 4-byte sync pattern and then by another 15 bytes of random data. These 34 bytes were transferred into PR4 by alternating signs in even and odd subsequences (note: the technique can certainly be applied also to EPR4 or E^2PR4, PR4 is given for illustration purpose only). Then, white gaussian noise was added.

At the decoding, the Viterbi decoding algorithm was applied followed by the postcoder $1 \oplus D^2$.

This gave a noisy version of the original sequence with an error rate .01 (extreme!).

Then, the 4-byte sync pattern was searched for within a Hamming distance of 4. That is, a window was slided, starting 8 bytes before and finishing 8 bytes after the beginning of the 4-byte sync pattern (note: corrective measures were not taken to prevent the occurrence of the sync pattern in this simulation).

Within the pattern at a distance of 4 or less, the first one giving the smallest distance, was taken. If no pattern at distance 4 or less was found, an uncorrectable error was declared (e.g., unable to achieve sync).

After 100,000 experiments, only in .128% of the cases the correct sync failed to be achieved. There were two cases of false sync (.002%), and, in .126% of the cases, sync was not achieved.

30,000 experiments were run under less extreme conditions, with a noise level of 10^{-5}, and in no case synchronization failed to be achieved.

Figure 6 illustrates a typical hardware configuration of an information handling/computer system in accordance with the invention and which preferably has at least one processor or central processing unit (CPU) 611.

The CPUs 611 are interconnected via a system bus 612 to a random access memory (RAM) 614, read-only memory (ROM) 616, input/output (I/O) adapter 618 (for connecting peripheral devices such as disk units 621 and tape drives 640 to the bus 612), user interface adapter 622 (for connecting a keyboard 624, mouse 626, speaker 628, microphone 632, and/or other user interface device to the bus 612), a communication adapter 634 for connecting an information handling system to a data processing network, the Internet, an Intranet, a personal area network (PAN), etc., and a display adapter 636 for connecting the bus 612 to a display device 638 and/or printer 639 (e.g., a digital printer or the like).

In addition to the hardware/software environment described above, a different aspect of the invention includes a computer-implemented method for performing the above method. As an example, this method may be implemented in the particular environment discussed above.

Such a method may be implemented, for example, by operating a computer, as embodied by a digital data processing apparatus, to execute a

sequence of machine-readable instructions. These instructions may reside in various types of signal-bearing media.

Thus, this aspect of the present invention is directed to a programmed product, comprising signal-bearing media tangibly embodying a program of machine-readable instructions executable by a digital data processor incorporating the CPU 611 and hardware above, to perform the method of the invention.

This signal-bearing media may include, for example, a RAM contained within the CPU 611, as represented by the fast-access storage for example. Alternatively, the instructions may be contained in another signal-bearing media, such as a magnetic data storage diskette 700 (Figure 7), directly or indirectly accessible by the CPU 611.

Whether contained in the diskette 700, the computer/CPU 611, or elsewhere, the instructions may be stored on a variety of machine-readable data storage media, such as DASD storage (e.g., a conventional "hard drive" or a RAID array), magnetic tape, electronic read-only memory (e.g., ROM, EPROM, or EEPROM), an optical storage device (e.g. CD-ROM, WORM, DVD, digital optical tape, etc.), paper "punch" cards, or other suitable signal-bearing media including transmission media such as digital and analog and communication links and wireless. In an illustrative embodiment of the invention, the machine-readable instructions may comprise software object code, compiled from a language such as "C", etc.

The present invention was described mainly as an alternative to the dual sync fields utilized in the prior art. However, it can be utilized in any system attempting to perform byte synchronization over random data. For instance, such an error tolerant pattern may be used following data to check if the system is in byte sync, otherwise the pattern is used for byte resynchronization.

While the invention has been described in terms of preferred embodiments, those skilled in the art will recognize that the invention can be practiced with modification within the spirit and scope of the appended claims.